Comparaison entre l'analyse logit et probit et les réseaux de neurones

Nicoleta Minoiu

Abstract. Cet article est une présentation comparative des performances de deux outils pour la fouille de données. Le premier est un outil statistique: le modèle logit ou probit. Le deuxiéme, les réseaux de neurones peut être aperçu comme un approximateur de fonctions universel. La première partie de l'article présente l'origine et le mode d'emploi des modèles logit et probit. La deuxième partie traite les réseaux de neurones et leurs propriétés. Enfin, les deux outils sont comparés d'un point de vue théorique et pratique par l'intermède d'un exemple fictif.

Mathematics Subject Classification 2000: 62-07, 68T10.

Key words: fouille de données, modèle logit, modèle probit, réseaux de neurones.

1 Motivation

Ces dernières années, pendant lesquelles les médias, les télécommunications et les technologies de l'information ont transformé notre société dans une société exclusivement basée sur l'information, on a constaté que le problème n'est pas d'obtenir et d'administrer les données, mais d'extraire les informations utiles à partir de ces données. De plus en plus nombreux sont les managers qui se voient confrontés avec le problème de ne pas pouvoir prendre une décision justifiée par une majorité des données disponibles, à cause de leures trop grandes dimensions.

Grace à la technologie moderne, de nos jours les données peuvent être mémorisées et traitées dans des bases de données d'une dimension variant de quelques gigaoctets à quelque terraoctets. La nécessité des mécanismes d'évaluation et de traitement automatique de ces bases de données a constitué le début d'une nouvelle science: Knowledge Discovery in Databases. Cette science utilise d'une part la technologie des bases de données et des outils statistiques mais aussi de l'intelligence artificielle ou l'apprentissage automatique.[1] (v. Figure 1).

Cet article a comme objectif une description comparative de deux outils qui peuvent être utilisés pour extraire des informations utiles à partir de très grandes bases de donnés: un outil statistique, le modèle logit ou probit, et un outil dérivé de l'intelligence artificielle, les réseaux de neurones. Le type de problème qui peut être résolu à l'aide des modèles logit ou probit est connu dans la littérature comme ,,Binary

Proceedings of The 2-nd International Colloquium of Mathematics in Engineering and Numerical Physics (MENP-2), April 22-27, 2002, University Politehnica of Bucharest, Romania.
BSG Proceedings 8, pp. 105-123, Geometry Balkan Press, 2003.

[1]V. Wiedmann/ Buckler (2001), p.21.

Choice-Model".[2] Un problème de type ,,Binary Choice-Model" peut être décrit de la manière suivante: un individu caractérisé par certaines propriétés doit faire un choix parmi deux types de comportement différent. Pour une population finie d'individus on connaît les propriétés, ainsi que le comportement adopté et on aimerait prédire le comportement pour un nouveau individu pour lequel on connaît uniquement les propriétés. Par exemple, l'individu pourrait être un client potentiel caracterisé par son age, son revenu et le nombre d'enfants. Si on dispose d'une base de données qui contient les caractéristiques de plusieurs individus, ainsi que leur décision d'acheter un produit ou pas, on pourra prédire non seulement la disposition d'un nouvel individu de devenir client, mais aussi quelles propriétés conditionnent la qualité de client.

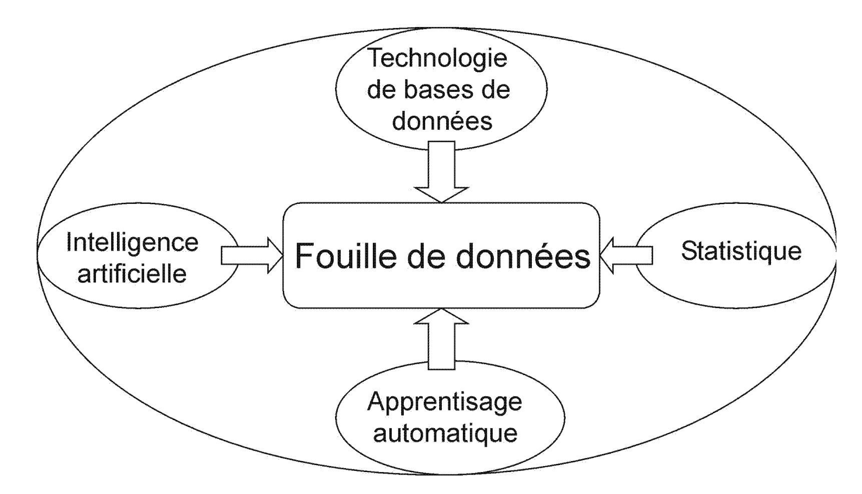


Figure 1: La fouille de données comme un domaine d'étude interdisciplinaire. Source: Nakhaeizadeh (1998), p. 2.

Dans l'article on décrit dans un premier temps les modèles logit et probit ainsi que les réseaux de neurones sans entrer dans les détails. Par la suite on fait une comparaison des deux outils d'un point de vue théorique et on insiste sur leur utilisation pratique sur un exemple. Les conclusions reflettent les résultats pratiques obtenus.

2 Les fonctions de répartition: Probit et Logit

,,Binary Choice-Model" est un problème de choix discret, dichotomique, qu'on peut trouver dans la littérature sous formes différentes. On va présenter ici trois façons différentes de l'aborder qui conduisent au même formalisme mathématique.

Le point de départ est à chaque fois est le procés aléatoire avec deux réalisations possibles, par exemple un procés de décision avec les valeurs symboliques "oui" et "non". Par définition, ces deux valeurs symboliques ont les valeurs entières "0" et

[2] V. Monfort (2000), p. 23 ff.

"1", ce qui nous permet d'introduire la variable aléatoire Y de la manière suivante: $Y = 1$ si la décision est "oui" et $Y = 0$ dans le cas contraire. Par la suite, on définit un vecteur de variables exogènes et mesurables qui conditionnent l'apparition de chacune des deux réalisations: $X = (X_1, X_2, \ldots, X_n)^T$, ainsi qu'un vecteur des coefficients ß=(ß$_1$, ß$_2$,... ß$_n$)^T.

Les trois façons d'aborder le modèle sont:

- Le procédé de la fonction d'utilité;
- Le procédé de la régression latente;
- Le procédé de l'expérance conditionnée.

Ces trois procédés vont être expliqués à l'aide des exemples concrets.

2.1 Le procédé de la fonction d'utilité

Ce type de ,,Binary Choice-Model" sert à prédire la décision d'un individu d'emmigrer ou pas. Ce problème peut être formalisé de la manière suivante: le vecteur X_0 décrit les conditions dans le pays d'emigration (température moyenne, densité de la population, le revenu moyen) et le vecteur X_1 décrit les mêmes conditions dans le pays d'immigration. Un individu a la fonction d'utilité $U_0 = \beta^T X_0 + \varepsilon_0$ pour le choix ,,0", de ne pas emmigrer, et l'utilité $U_1 = \beta^T X_1 + \varepsilon_1$ pour le choix ,,1", emmigration. ε_0 si ε_1 sont des variabiles aléatoires, composantes de l'utilité individuelle, qui n'apparaissent pas dans le vecteur X. L'individu choisit l'alternative "1" lorsque l'utilité U_1 est supérieure à l'utilité U_0.

Soient $\varepsilon = \varepsilon_0 - \varepsilon_1$ et $\beta^T X = \beta^T X_1 - \beta^T X_0$. Le comportament peut être décrit mathématiquement par l'expression suivante:

$$U_1 > U_0 \quad \Rightarrow \varepsilon_0 - \varepsilon_1 \leq \beta'^T X_1 - \beta'^T X_0 \quad \Rightarrow \varepsilon \leq \beta'^T X.$$

La probabilité du choix ,,1" est ainsi la probabilitaté que l'utilité U_1 soit supérieure à l'utilité U_0[3]:

$$W(Y = 1 \mid X) = W(U_1 > U_0) = W(\varepsilon_0 - \varepsilon_1 \leq \beta'^T X_1 - \beta'^T X_0) = W(\varepsilon \leq \beta'^T X)$$

2.2 La regression avec une variable latente[4]

Pour cette modélsation on fait l'hypothèse de l'existence d'une variable latente Y_i^* , qui représente une combinaison des caracteristiques X_i d'un individu. Par exemple, pour des problèmes de type ,,credit scoring" Y_i^* peut être la bonité d'une entreprise ,,i" et Y_i la décision oui ou non d'accorder un crédit. Y_i^* est ensuite décrit par une regression linéaire $Y_i^* = \beta'^T X_i + \varepsilon_i$. ε_i est une variable aléatoire qui représente les influences non-negligeables mais aussi non-mesurables du milieu sur la variable Y_i^*. La variable aléatoire Y_i est définie par la formule suivante:

$Y_i = 1$ pour $Y_i^* > 0$

[3]V. Langche Zeng ,,Prediction and Clasification with Neural Network Models", p. 4

[4]v. Alain Monfort ,,Statistique", p. 23.

$Y_i = 0$ pour $Y_i^* < 0$.
De cette façon la probabilité d'une décision positive ($Y_i = 1$) est égale à la probabilité d'une bonité positive $Y_i^* > 0$ et après les calculs on obtient la même formule que pour le procédé de la fonction d'utilité:

$$W(Y_i = 1 \mid X_i) = W(Y_i^* > 0) = W(\varepsilon_i > -ß^T X) = W(\varepsilon_i < ß^T X)$$

2.3 Le procédé de l'espérance conditionnée

Ce type de ,,Binary Choice-Model" définit Y comme une variable aléatoire discrète et binaire, qui peut prendre les valeurs ,,0" et ,,1". La probabilité de l'évenement $Y = 1$ est exprimée à l'aide d'une fonction inconnue $F(X, ß)$, qui doit avoir les propriétés d'une fonction de répartition[5]. La distribution de la variable aléatoire Y est la suivante:

$$W(Y = 1) = F(X, ß) \text{ und } W(Y = 0) = 1 - F(X, ß).$$

L'espèrance conditionnée de la variable aléatoire Y, sachant X va être alors:

$$E[Y|X] = 0 * [1 - F(X, ß)] + 1 * [F(X, ß)] = F(X, ß).$$

Si pour les deux premiers procédés on suppose F comme fonction de répartition des variables aléatoires ε , et ε_i, alors on va voir que dans les trois types de ,,Binary Choice-Model" la fonction de répartition de la variable aléatoire conditionnée Y est la suivante:

$$W(Y = 1|X) = F(\beta^T X) \text{ und } W(Y = 0|X) = 1 - F(\beta^T X).$$

Pour les deux premiers cas on a supposé une combinaison linéaire $\beta^T X$. Cette hypothèese peut être valable également pour le troisième cas, tant que F a les propriétés d'une fonction de répartition.

2.4 Les distributions probit et logit

Il est évident que pour les trois types de ,,Binary Choice-Model" la distribution de la variable dépendante Y est détéminée par la distribution de la variable ε. On se demande alors quelle serait cette distribution. Si les effets de plusieurs influences extérieures sont superposés, le choix d'une distribution gaussienne pour ε serait justifié par le théorème limite centrale[6]. Le modèle Probit est définit de cette façon:

$$W(Y = 1\,|X\,) = F(\beta'^T X) = \int_{-\infty}^{\beta'^T X} \varphi(t)dt \quad \text{avec} \quad \varphi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}.$$

[5] v. Greene (1993), p. 813 ff.
[6] v. Theil (1971), p. 630

L'hypothèse d'une distribution normale standard pour ε ne change pas la structure du modèle, car toute variable avec une distribution normale peut être ramenée à une distribution normale standard, de moyenne nulle et variance 1[7].

Pour beaucoup d'applications on utilise la distribution logit:

$$W(Y = 1\,|X\,) = F(\beta^T X) = \frac{e^{\beta'^T X}}{1 + e^{\beta'^T X}}.$$

La différence de cette distribution par rapport à la distribution probit est que la fonction F varie plus vite autour de $\beta'^T X$=0 (v. Figure 4). Le choix parmi les deux distributions est difficilement justifiable d'un point de vue théorique et dépend de l'application. Pour effectuer les calculs la distribution logit semble plus avantageuse, cependant dans la plupart des applications il n'y a pas une différence notable de performance[8]. L'estimation des coefficients ß se fait par la méthode de maximum de vraisamblance (Maximum Likelihood).

3 Les réseaux de neurones

Si les analyses probit et logit sont des procédés économetriques caractérisés par deux étapes (la création d'un modèle suivie par l'estimation de ses paramètres), les réseaux de neurones appartiennent à une catégorie différente d'outils d'analyse des données.

Comme leur nom le suggére, les réseaux de neurones ont eu comme point de départ les connaissances biologiques et plus précisement neuro-physiologiques à propos du cerveau humain. Les réseaux de neurones biologiques sont des ensembles de neurones qui amplifient ou attenuent les signaux qui traversent leurs liaisons. Un neurone est constitué d'un noyau, de dendrytes qui reçoivent le signal d'entrée, et l'axon. La communication entre les neurones est de nature électrochimique et elle est assurée par des synapses. Les réseaux de neurones artificiels sont un modèle symplifié du mode de fonctionnement des réseaux biologiques décrits plus haut. L'objectif est de créer des systèmes qui ont la plus importante propriété du cerveau humain, la capacité d'apprentissage. En effet, on peut dire qu'après un processus de préparation les réseaux de neurones artificiels apprennent un certain comportement. Un réseau de neurones peut être appris à distinguer les potentiels clients des personnes non interessées, à partir d'un échantillon représentatif d'individus. Comment cela peut être possible, quelle est la structure d'un réseau de neurones et par quel moyen l'apprentissage devient possible va être décrit dans les paragraphes suivantes.

3.1 Définitions

D'un point de vue global, on peut regarder les réseaux de neurones comme des boîtes noires avec au moins une entrée et une ou plusieurs sorties. A l'intérieur de ces boîtes

[7] V. Greene (1993), p. 819
[8] V. Greene (1993), p. 815

il y a des neurones qui jouent le rôle d'opérateurs de calcul et des connexions entre eux.

Par définition[9] un neuron n_i, est caracterisé à l'instant t par le touple

$$(X(t), W_i(t), a_i(t), f, g, h).$$

Dans ce touple on a:

$X(t) = (x_1(t), x_2(t), ..., x_n(t)) \in \mathbb{R}^n$ le vecteur d'entrée à l'instant t,

$W_i(t) = (w_{i1}(t), w_{i2}(t), ..., w_{in}(t)) \in \mathbb{R}^n$ le vecteur des poids à l'instant t,

$a_i(t) \in R$ l'état d'activation du neurone à l'instant t,

$h : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ avec $s_i(t) = h(X(t), W_i(t))$ la fonction de propagation, qui génère le signal d'entrée$s_i(t)$,

$g : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ avec $a_i(t) = g(s_i(t), a_i(t-1))$ la fonction d'activation, qui calcule l'état d'activation $a_i(t)$ à l'instant t et

$f : \mathbb{R} \to \mathbb{R}$ cu $y_i(t) = f(a_i(t))$ fonction de sortie, qui donne la sortie $y_i(t)$ du neurone i à l'instant t.

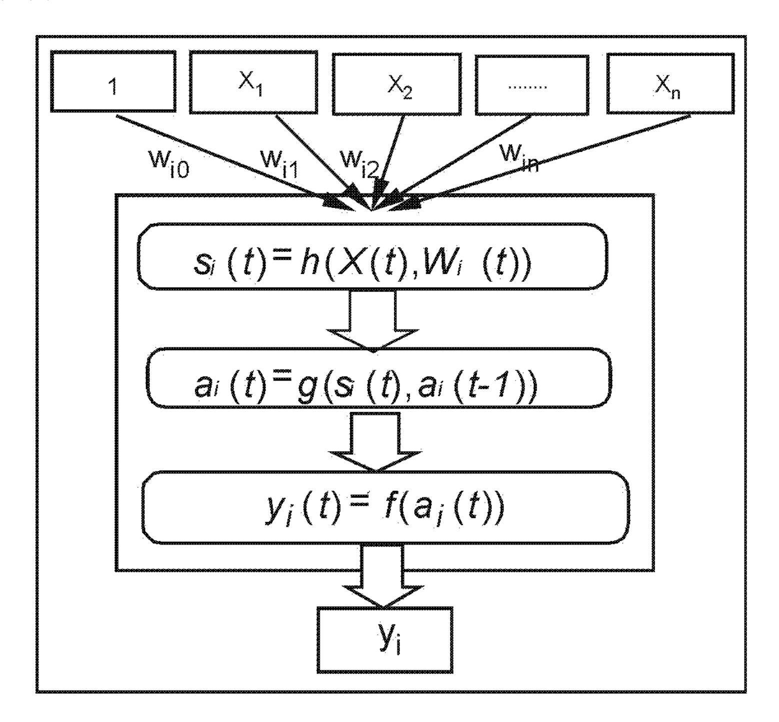


Figure 2: Construction d'un neurone. Source: Scherer (1997), p. 46.

[9]V. Hagen (1997), p. 6 f.

Mais qu'est-ce que représentent toutes ces fonctions et vecteurs (Figure 2)[10] ? Le vecteur d'entrée $x(t)$ peut représenter les valeurs d'entrée données par l'utilisateur, pour la première couche, ou les sorties de la couche antérieure pour les autres couches de neurones. Si le neurone est situé dans la première couche, une des entrées doit être constante. Les poids $W_i(t)$ qui vont être modifié pendant l'apprentissage[11] modélisent la plasticité synaptique. L'état d'activation définit l'état actuel d'un neurone. La fonction de propagation décrit le traitement de l'information qui est fait par le neurone. La fonction d'activation calcule la transformation de l'état d'activation à l'instant t+1 à partir de l'état d'activation à l'instant t. La fonction de sortie calcule la valeur de sortie d'un neurone en fonction de sont état d'activation.

Les neurones sont les composantes de base d'un réseau de neurones. Par la connexion des sorties et des entrées on crée des structures comme présentées par exemple sur la Figure 5 dans l'annexe. En principe on peut connecter des neurones de types différents, mais en pratique on utilise au plus deux types de neurones[12].

Dans la plupart des cas les fonctions d'activation des neurones ne sont pas linéaires. On peut remarquer facilement une forte non-linéarité des sorties, même si les fonctions de propagation dans la couche de sortie sont linéaires. La connexion des neurones dans plusieurs couches permet de modéliser par le réseau des fonctions très complexes.

Le choix de la fonction d'activation dépend du type de l'application. Dans nombre d'application on utilise des sigmoïdes comme fonctions d'activation. D'autres exemples sont la fonction logit ou la tangente hyperbolique. Un réseau de neurones artificiels avec des fonctions d'activation sigmoïdales et plusieurs couches s'appelle perceptron multi-couche (Multilayer-Perzeptron - MLP). Pour des classifications (clustering), on utilise une autre classe de fonctions, des fonctions avec activation locale. Les réseaux de neurones artificiels avec de telles fonctions d'activation s'appellent Radial Basis Function – RBF. La fonction identité ou la fonction marche peuvent aussi être employées comme fonctions d'activation[13].

La topologie est la façon de connecter les neurones dans un réseau de neurones. On a par exemple la topologie Feedforward (les neurones transmettent les signaux dans une seule direction) et la topologie Feedback (qui permet l'existence des boucles)[14]. Des liaisons bidirectionnelles sont aussi possibles.

3.2 Procédé d'apprentissage: l'algorithme de rétropropagation

Après les définitions mathématiques des réseaux de neurones, il est légitime de se demander pourquoi ces structures compliquées trouvent de plus en plus d'utilisations ces dernières années. La réponse est donnée par une démonstration du mathématicien russe Kolmogorov, qui a prouvé en 1957 que toute fonction continue peut être représentée

[10] Ausführlicher: Scherer (1997), p. 47 ff. und Hagen (1997), p. 7 ff.
[11] pour la définition du processus d'apprentisage v. Kapitel 0.
[12] V. Adamy (2000), p. 128.
[13] v. Hagen (1997), p. 12.
[14] Plus de détails: Scherer (1997), p. 54 ff.

par un réseau avec un nombre fini de neurones avec n'importe quelle précision[15].

Mais comment est-il possible pour un réseau d'approximer toute fonction continue? Ce processus est appellé apprentissage. Le succés des réseaux de neurones est dû au fait qu'elles sont capables d'apprendre un certain comportement à partir des données exemple. De ce point de vue, les réseaux de neurones artificiels ont un comportement similaire aux réseaux biologiques.

L'apprentissage à l'aide des données exemple peut être surveillée, quand les sorties du réseau sont comparées aux sorties exemple, ou non surveillée. Dans ce deuxième cas on n'utilise pas des sorties exemple et les poids sont appellés des fonctions énergétiques[16].

Un procédé de calibration du réseau de neurones est une méthode de calcul de poids optimaux[17]. L'algorithme le plus utilisé dans ce but est l'algorithme de rétropropagation, Backpropagation, qui peut être utilisé pour l'apprentissage surveillé.

L'apprentissage surveillé est un apprentissage par correction des erreurs. C'est pour quoi il faut d'abord définir une fonction d'erreur. Cette fonction pourrait être par exemple l'erreur moyenne quadratique $\varepsilon = \frac{1}{2s}\sum_{i=1}^{s}\sum_{j=1}^{m}(Y_{i,j}^{-}Y_{i,j}^{*})^2$ où s représente le nombre de couples $(X_i^{;}Y_i)$ utilisés pour l'apprentissage, m est le nombre de neurones dans la couche de sortie et Y_i^* est la réponse du réseau pour l'entrée X_i. L'algorithme de rétropropagation peut être appliqué pour n'import quel type de fonction d'erreur.

La fonction d'erreur dépend de tous le poids du réseau et doit être minimisée. A cause de la non-linéairité du réseau on ne peut pas trouver un minimum gloabal par une méthode analytique. C'est pour cette raison qu'on cherche un minimum local par une méthode itérative par une descente de gradient récursive. La méthode est décrite dans tous les détails dans Hagen[18].

Une remarque intéressante est que la solution finale de ce processus d'optimisation, représentée par les valeurs des poids en fin d'apprentissage, peut être différente d'un apprentissage à l'autre même si les donées utilisées pour la calibration sont les mêmes. Ceci car cette solution dépend des conditions initiales, comme tout procédé d'optimisation locale.

Les réseaux de neurones sont utilisés pour des application où la fonction qui lie les entrées et les sorties est inconuue. On dispose uniquement d'un certain nombre de combinaisons d'entrées et de leurs sorties correspondantes. Après la calibration du réseau à partir des données exemple, il est important de vérifier les performances du réseau sur de nouvelles données. En effet, l'objectif est d'apprendre un comportement général et non pas de réproduire les données exemple, propriété des réseaux de neurones appellée capacité de généralisation. Par capacité de généralisation on entend que les réseaux peuvent interpoler ou extrapoler avec une bonne précision, après un apprentissage correct, et même si les données d'entrée ne sont plus celles

[15]v. Wiedmann/Buckler (2001), p. 59.
[16]v. Hagen (1997), p. 19 ff.
[17]v. Wiedmann/Buckler (2001), p. 53.
[18]V. Hagen (1997), p. 24 ff.

de la base d'apprentissage le réseau calcule correctement les sorties correspondantes. Pour améliorer cette capacité de généralisation, les données doivent être choisies de manière qu'elles soient représentatives pour le domaine étudié.

Malheureusement, dans les applications réelles on dispose de données affectées par le bruit, distorsionnées. Dans ce cas-là, le réseau risque de s'adapter trop aux données et d'apprendre également le bruit. Ce phénomène s'appelle sur-adaptation ou sur-généralisation[19]. Pour éviter ce phénomène, les données disponibles sont divisées dans trois ensembles: des données d'apprentissage, des données de validation et des données de test. Les données d'apprentissage servent à la calibration des poids du réseau, modifiés en fonction de l'erreur effectuée. Pendant l'apprentissage on calcule également l'erreur sur les données de validation. Cette erreur n'est pas utilisée pour modifier les poids du réseau, mais pour détecter les sur-apprentissage. Au début, l'erreur sur les données de validation doit diminuer, le réseau apprend la fonction sous-jacente. Si cette erreur augmente, le réseau commence à apprendre le bruit et il a un mauvais comportement en généralisation. Les données de test sont utilisées pour évaluer les performances du réseau en fin d'apprentissage.

A la fin de ce paragraphe dédié au procédé d'apprentissage par rétropropagation il faut accentuer sur le fait que ce n'est qu'un algorithme d'optimisation. Ceci signifie que le procédé ne trouve pas un optimum global, car on ne sait pas éviter les minimums locales ou les surfaces planes de la fonction d'erreur.

3.3 Les proprétés des réseaux de neurones

Comme vu plus haut, les réseaux de neurones peuvent être assimilés dans certaines conditions à des approximateurs universels. Ce fait explique pour quoi les réseaux de neurones peuvent prédire un comportement décisional et peuvent résoudre de problèmes de type "Binary Choice Model": les réseaux de neurones artificiels peuvent approximer la relation entre les caractéristiques d'un individu et son comportement décisional. En résumé, ceci est possible à cause de ces quatre propriétés des réseaux de neurones[20]:

Non-linéarité: la relation recherchée ne doit pas être linéaire, les dépendaces non-linéaires sont même mieux approximées.

Capacité d'apprentissage: Il faut faire aucune hypothèse à propos de la forme de la dépendance, celle ci va être déterminée à partir des données exemple.

Capacitaté de généralisation: Même si les données d'appprentissage sont bruitées, le réseau peut apprendre le processus initial.

Nombre de variables: Le nombre de variables d'entrée n'est pas limité comme pour d'autres procédés d'interpolation, par exemple l'interpolation spline.

Il y aussi des propriétés moins souhaitables, comme par exemple le fait que la fonction reste inconnnue à l'utilisateur, qui a accès seulement aux sorties qui correspondent à des entrées données. On va revenir sur ces propriétés dans un paragraphe

[19] V. Wiedmann/Buckler (2001), p. 62.
[20] V. Wiedmann/Buckler (2001), p. 45. Adamy (2000), p. 120 ff. Zeng (1996), p. 8 ff.

ultérieur, lorsqu'on va comparer les réseaux de neurones avec les modèles logit et probit.

4 Les modèles probit et logit par rapport aux réseaux de neurones

Après avoir présenté les modèles logit et probit et le réseaux de neurones comme des méthodes appropriées pour „Binary Choice Model", on va faire une comparaison des deux outils de cette nouvelle science, Knowledge Discovery in Databases. On va essayer de répondre aux questions suivantes:

Quelles sont les similarités et le différences de deux outils?

Quels sont leurs avantages et désavantages respectifs?

La réponse à ces questions commence avec une comparaison d'un point de vue mathématique et continue avec une présentation comparative des plusieurs propriétés qui peuvent influencer la décision d'utiliser ces outils pour des problèmes pratiques: non-linéarité, capacité d'apprentissage, capacité de généralisation, complexité du modèle, conditions imposées aux données disponibles, l'interpretation des résultats.

4.1 Comparaison d'un point de vue mathématique

Les réseaux de neurones et les modèles logit et probit sont similaires d'un point de vue mathématique. Un réseau de neurones avec un seul neurone et une fonction d'activation probit ou logit ont la même représentation mathématique que le modèle probit et logit respectivement (v. Figure 3). Les réseaux de neurones peuvent être vus comme des modèles probit et logit non-linéaires, avec des relations complexes entre les variables endogènes[21].

Une différence entre les deux méthodes est le procédé de calcul des coeffiecients (des poids): la méthode de maximum de vraisemblance pour les modèles logit et probit et l'algorithme de rétropropagation pour les réseaux de neurones. La méthode de maximum de vraisemblance est un procédé analytique qui conduit à un minimum global. L'algorithme de rétropropagation est au contraire un procédé d'optimisation qui ne garantit qu'un minimum local. En plus, le temps de calcul de cet algortihme est beaucoup plus élevé.

Une autre différence est la signification des valeurs de sortie des deux méthodes. Les méthodes logit et probit donnent la probabilité d'une décision positive, tandis que les réseaux de neurones sont calibrés pour des réponses binaires et non pas pour une probabilité. Ceci car les données exemple appries par le réseau ont que des sorties „0" ou „1" et le réseau apprend un comportement dont la réponse soit une de ces deux valeurs discrètes.

[21] V. Sarle (1994), p. 3 ff.

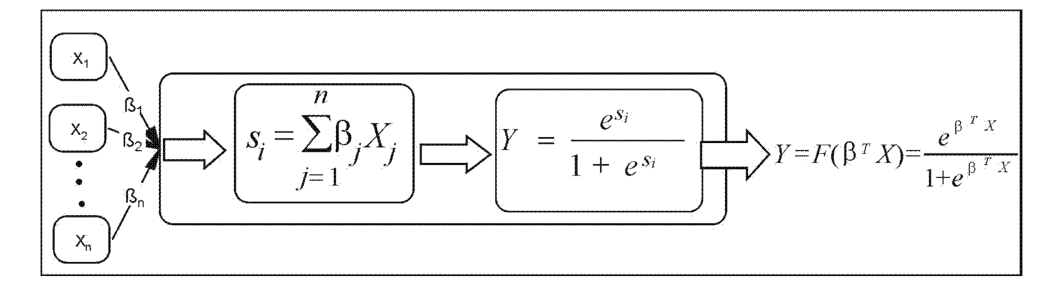


Figure 3: Un réseau avec un neurone et le modèle logit.

4.2 La non-linéarité

La non-linéarité des réseaux de neurones leur offre un grand avantage par rapport aux modèles probit ou logit. Les réseaux de neurones peuvent saisir toute sorte d'interactions entre les variables endogènes pour calculer la sortie. Les modèles logit et probit supposent une dépendance linéaire entre les variables endogènes et la variable latente. D'un point de vue théorique, le réseau de neurones peut modéliser toute relation représentée par les données exemple.

4.3 La capacitaté d'apprendre

La capacité d'apprendre représente pour les réseaux de neurones une caractéristique fondamentale. Pendant le processus d'apprentissage les réseaux de neurones ajustent dynamiquement les poids et éventuellement les paramètres des fonctions d'activation, tels que les données d'apprentissage soient réproduites avec la meilleure précision possible. Il y a aucune restriction pour la relation entre les variables d'entrée et les sorties. Au contraire, pour les modèles probit et logit l'utilisateur doit imposer à priori la forme de la dépendance, ce qui est plus délicat.

Pour l'estimation des modèles logit ou probit on calcule les coefficients ß mais pas la forme fonctionnelle. Le chercheur peut utiliser plusieurs hypothèses, il peut ajouter ou éliminer les variables, superposer le modèles avec les données empiriques, mais malgré toutes ces possibilités les procédés statistiques restent des modèles statiques qui n'ont pas la capacité intrinséque d'apprendre et de s'adapter automatiquement à l'application pratique.

4.4 La complexité du modèle

La modélisation peut représenter une activité compliquée pour les réseaux de neurones. Si on veut de très bonnes précisions, il faut adapter le réseau au problème pratique. Par la modélisation du réseau on entend choisir les fonctions de propagation, les fonctions d'activation et de sorties des neurones, choisir le nombre de couches et le nombre de neurones sur chaque couche. Le meilleur choix peut être fait par des

essais empiriques, mais il existe aussi des techniques d'optimisation[22]. Ces techniques permettent d'obtenir des réseaux très spécialisés pour l'application. Ceci peut être un avantage par rapport aux modèles probit et logit, mais comme presque toute application a besoin d'une architecture dédiée, les outils d'optimisation sont très couteux en temps de calcul.

4.5 Interprétation des résultats

La possibilité d'interpréter les résultats obtenus après l'évaluation d'une base de données est très importante pour la comparaison des deux outils. Comme on l'a vu plus haut, les réseaux de neurones ont en désavantage considérable: ils restent pour l'utilisateur un modèle boîte noire. Les dépendances entre les variables endogènes et exogènes vont rester inconnues à l'utilisateur à cause de la forme complexe du modèle. Les effet marginales des variables d'entrée sont eux aussi difficiles à calculer.

Du point de vue des possibilités d'interprétation, les modèles logit et probit sont supérieurs aux réseaux de neurones. On peut par exemple calculer à l'aide d'une dérivé partielle de la variable latente par rapport à une variable exogène l'effet marginal de cette variable. L'effet marginale estimé de la variable X_j est exactement le coefficient qui lui correspond, $\tilde{\beta}_{j.}$

$$Y^* = \beta^T X_j + \varepsilon_j \Rightarrow \frac{\partial Y^*}{\partial X_j} = \tilde{\beta}_j$$

De l'autre côté, le signe du coefficient $\tilde{\beta}_j$ estimé montre si l'influence de la variable latente est croissante ou décroissante par rapport à la variable X_j.

Malgré ces difficultés, il y a quelque possibilités d'intepréter les résultats des réseaux de neurones, par exemple les procédés de Input- Pruning [23]. Un procédé de Input- Pruning est une technique d'optimisation de la complexité du réseau de neurones par l'élimination des variables d'entrée. Cette technique permet de savoir si une variable d'entrée a une influence sur la sortie ou pas.

4.6 La capacitaté de généralisation

La capacité de généralisation caractérise les réseaux de neurones et également les modèles logit et probit. La question reste d'évaluer les performances d'interpolation et d'extrapolation des deux outils. A cause de la structure compliquée des réseaux de neurones la réponse à cette question ne peut être donnée que par des études empiriques, les possibilités des études théoriques sont trop limitées face à cette complexité.

Pour effectuer la comparaison on a utilisé plusieurs critères qui ont à la base le nombre de décisions estimées correctement et le nombre total de données de test. Le taux de succées est calculé comme poucentage des décisions estimées correctement[24].

[22] V. Hertz/Krogh/Palmer (1992), p. 156.
[23] V. Wiedmann/Buckler (2001), p. 71.
[24] V. Zeng (1996), p. 7.

Comme une constatation générale des études empiriques, on peut dire que les performances des réseaux de neurones sont meilleures que celle des modèles probit et logit, mais cependant très variables en rapport avec le type d'application. Dans les application où les performances des deux outils sont comparables, on a une dépendance linéaire entre les variables exogènes et la variable latente. C'est le cas de l'étude réalisée par Ainslie et Dreze[25], dans laquelle on cherche à prédire la décision d'acheter une certaine marque d'automobile en fonction des caractéristiques individuelles: „Intercept, Asset level, Income, Credit Card, Upscate Retail Card, Financial Distress und Auto Loan".

4.7 La qualité et la quantité des données

L'analyse de l'influence de la qualité et de la quantité des données sur les résultats obtenus peut être faite aussi uniquement par des méthodes empiriques. Langche Zeng a montré dans une de ses études, „Prediction and Classification with Neural Network Models", que jusqu'à un certain niveau du bruit, les réseaux de neurones ont des meilleurs perfromances que les modèles probit et logit. Mais même les réseaux de neurones ont un taux d'erreur élevé pour de grands niveaux du bruit.

Dans le chapitre suivant de cet article on va présenter une étude empirique de la dépendance entre le taux de succés et le niveau de bruit dans le cas d'un modèle linéaire et d'un modèle non-linéaire estimés à l'aide des deux outils.

5 Le modèle probit et les réseaux de neurones: comparaison pratique

Comme décrit au chapitre précédent, la capacité de prédiction des modèles probit, logit et réseaux de neurones est influencée de manière significative par le type de relation entre les variables exogènes et la variable latente, ainsi que par la qualité des données. Par la suite on va montrer par un exemple fictif que, malgré les meilleurs performances des réseaux de neurones par rapport aux modèles logit et probit, leur capacité de généralisation est affectée par le niveau du bruit,

5.1 La construction de l'exemple fictif

L'exemple fictif sur lequel on va tester les deux outils (les réseaux de neurones et les modèles logit et probit) a été constitué de la manière suivante. On disposait d'une base de données qui contient 600 observations, soit l'age et le revenu de 600 individus différents. Le revenu X_{i1} (v. Figure 7, Annexe) et l'age X_{i2} (v. Figure 8, Annexe) d'un individu „i" ont été supposées des variables exogènes. La variable latente Y_i^* a été constituée par la relation non-linéaire suivante entre les variables exogènes:

$$Y_i^* = \beta_1 \cdot X_{i1} + \beta_2 \cdot X_{i1} \cdot X_{i2} + \beta_3 + \varepsilon_i.$$

[25] V. Ainslie/Dreze (1996), p. 9-10

On a choisi pour les coefficients β_1, β_2 la valeur 1 et pour β_3 la valeur „-414“. Avec ces coefficients, la variable latente a la forme suivante:

$$Y_i^* = X_{i1} + X_{i1} \cdot X_{i2} - 414 + \varepsilon_i \quad .$$

Dans cette dernière expression, ε_i est un bruit gaussien, de moyenne nulle et de variances différentes: $\varepsilon_i \sim N(0, \sigma)$. La variable aléatoire Y est définie par:
$Y_i = 1$ pour $Y_i^* > 0$
$Y_i = 0$ pour $Y_i^* \leq 0$
La valeur du coefficient β_3 („-414“) a été la valeur médiane de $X_{i1} + X_{i1} \cdot X_{i2}$, dans le but d'avoir dans l'échantillon disponible un nombre égal de décisions positives et négatives pour une variable latente sans bruit.

La variance du bruit ε_i a été respectivement 0, 5%, 10%, 15% et 25% de la valeur médiane. Plus la variance est élevée, plus la probabilité des grandes valeurs du bruit augmente.

5.2 Utilistion concrète des deux outils

Pour la calibration de deux outils on a utilisé 300 observation de l'échantillon disponible. On a calculé ensuite le taux de succés du modèle probit pour les données utilisées dans la calibration, ainsi que pour les autres 300 observations. Pour les réseaux de neurones ces dernières 300 observations ont été divisées dans deux parties: une partie (35) ont été utilisées comme données de validation, c'est à dire comme témoin pour arrêter le processus d'apprentissage, et sur le reste on a calculé le taux de succés de la prédiction du réseau.

Pour estimer le modèle probit on a utilisé Software Limdep, Version 7.0 , Econometric Software (written by William H. Greene). Ce logiciel fait une estimation de la variable latente d'après l'expression suivante:

$$Y_i^* = \beta'^T X_i + \varepsilon_i.$$

Si on introduit pas de manière explicite une constante comme variable endogène pour le coefficient β_3, alors ce logiciel ne prend pas en compte ce coeffcient.

D'après les variables endogènes utilisées on peut distinguer trois cas essayés de manière empirique dans cet exemple:

1. Les variabile endogènes sont X_{i1},X_{i2} et la constante „1“ qui correspond au terme libre β_3;
2. Les variabile endogènes sont X_{i1},$X_{i1} \cdot X_{i2}$ et la constante „1”;
3. Les variabile endogènes sont X_{i1},X_{i2}.

Pour chaque paire (X_{i1},X_{i2}), Software LimDep a reçu la valeur correspondante de la variable aléatoire Y_i.

Les sorties fournies par le programme ont été les coefficients des variables endogènes et le taux de succés pour les 300 observations utilisées pour l'estimation. Le taux de succés pour les autres 300 observations a été calculé en Excel.

Pour calibrer les réseaux de neurones on a utilisé le logiciel ,,Neural Connections" prouduit par Recognition System Ltd. On a choisi un réseau avec une topologie simple: un réseau feedforward avec deux neurones dans la couche d'entrée, une couche cachée avec deux neurones et un neurone de sortie. Les fonctions d'activation de tous les neurones ont été des sigmoïdes. La méthode de calibration utilisée a été la méthode du gradient conjugué, une extension de l'algorithme de rétropropagation. Après le réseau a été calibré avec les 300 observations utilisées pour la calibration du modèle probit, après il a été validé par 35 des observations restantes, le logiciel nous a fourni le taux de succés sur le reste de 265 d'observations.

Les résultats du modèle probit et du réseaux de neurones sont montrés sur la Figure 6, dans l'Annexe.

5.3 Interpretation de résultats

On va d'abord interpréter les résultats de deux cas pour le modèle probit pour lequel on a supposé l'existence d'un coefficient libre β_3. Les taux de succés pour les observations utilisées pour la calibration sont comparables et supérieurs à 90%, même pour des niveaux de bruit élevés. Pour les observations qui n'ont pas été utilisées pour la calibration sont au contraire très différents. Dans le cas où on a utilisé X_{i1},X_{i2} comme variables exogènes, le taux de succées très proche de la valeur minimale adimissible, 50%. D'autre part, si on utilise X_{i1} et $X_{i1} \cdot X_{i2}$ comme variables exogènes, le taux de succés sur l'ensemble de calibration est comparable au taux de succés sur l'ensemble de test. Ce comportement du modèle probit montre que sa capacité de prédiction est très réduite dans les cas non-linéaires. Le niveau du bruit semble ne pas avoir une grosse influence sur le taux de succés.

Le troisiéme cas du modèle probit, où les variables exogènes ont été X_{i1}, X_{i2} va être comparé par la suite avec le réseau de neurones qui a mêmes variables d'entrée, X_{i1},X_{i2}. Le taux de succés pour les observations utilisées dans le processus d'estimation du modèle est considérablement inférieure à celui du réseau de neurones calculé pour les mêmes données. La sensibilité du réseau de neurones par rapport au niveau du bruit est par ailleurs supérieure au modèle probit. Pour un très grand niveau du bruit ($\sigma = 103,5$), les performances des deux méthodes restent à 70%.

On peut dire que pour cette application concrète la capacité de prédiction des réseaux de neurones est beaucoup supérieure par rapport au modèle probit, mais seulement pour un niveau de bruit relativement faible.

Dans le cas du modèle probit on peut interpréter les valeurs des coefficients estimés et on peut ainsi trouver la forme estimée de la variable latente et donc la façon don't les variables d'entrée influent sur la sortie. Par exemple on porrait trouver qu'une augmentation du revenu génère une hausse de la variable latente Y_i^* et implicitement la probabilité d'une décision favorable. Un augmentation de l'age de l'individu a au contraire un effet négatif.

6 Conclusions

En fin, après une comparaison théorique et pratique des deux outils on tempte de donner une réponse à la question: lequel des deux outils devrait être utilisé pour un problème concrét ?

A mon avis, avant de se poser cette question il faut décider le but de l'analyse: une bonne prédiction ou la compréhension des influences de différentes variables ?

Comme décrit plus haut, pour les réseaux de neurones il y a un conflit entre la qualité de la prédiction et l'interprétabilité des coefficients du modèle. Si la transparence du modèle n'est pas une priorité, les réseaux de neurones peuvent être utilisés avec succés pour des fonctions de prédiction. Les autres désavantages des réseaux de neurones (le temps de calcul, la complexité du modèle) peuvent être améliorés par des automatiosations.

Si au contraire la possibilité de comprendre les relations inclues dans le modèle est importante, on peut toujour bénéficier des avantages des réseaux de neurones, en utilisant les deux outils en parallélle. Les réseaux de neurones font des bonnes prédictions et les modèles logit et probit servent pour l'interprétation des relations entre les variables exogènes et les variables latentes. Dans ce cas-là, les réseaux de neurones servent de Benchmark-Modell[26]

Les réseaux de neurones sont à mon avis un extension très utile des modèles économétriques conventionnels, mais ils ne peuvent pas les remplacer. Ces méthodes traditionnelles sont préférables lorsqu'on connaît les dépendences de causalité entre les variables. Dans ce cas-là elles peuvent donner des meilleurs résultats que les réseaux de neurones.

7 Annexe

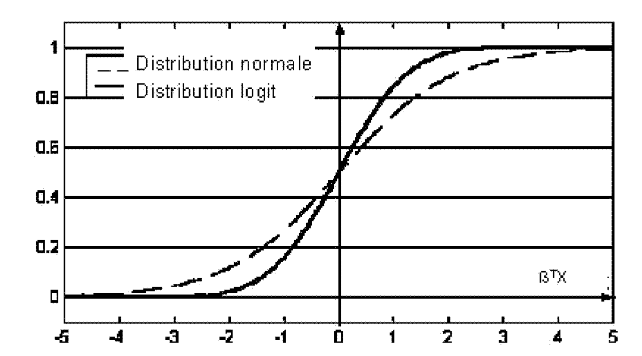


Figure 4: Distribution normale et distribution logit.

[26]V. Ainslie/Dreze (1996), p. 12.

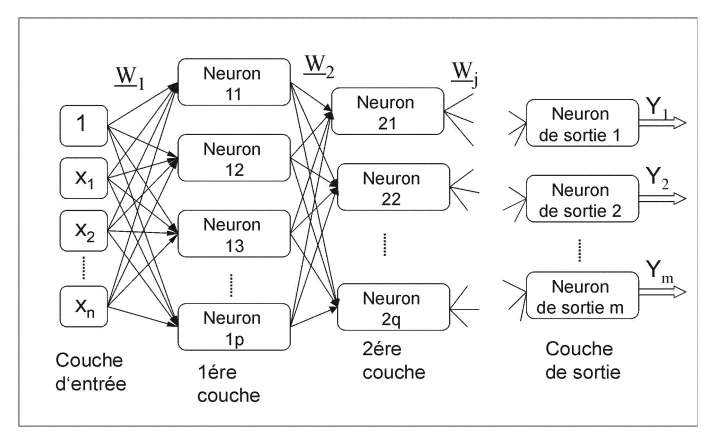


Figure 5: Un réseau de neurones.

	Le modèle probit Y=ß1*x1+ß1*x1*x2+ß3+e Les variables exogènes :X1;X2			Le modèle probit Y=ß1*x1+ß1*x1*x2+ß3+e Les variables exogènes :X1;X1*X2		
La variance du bruit e:N(0,si)	Les coefficients estimés	Le taux de succés sur les données de calibration	Le taux de succés sur les données de test	Les coeffcients estimés	Le taux de succés sur les données de calibration	Le taux de succés sur les données de test
0	ß1=1,960731903 ß2=0,4695876626 ß3=-39,97565339	97,32%	52,64%	ß1=2,012093814 ß2=1,703118524 ß3=-723,1222425	100,00%	99,33%
20,7	ß1=1,155101623 ß2=0,3068662400 ß3=-24,52296512	96,32%	52,98%	ß1=-0,0564588581 ß2=00,04382100361 ß3=-16,88090237	97,66%	97,01%
41,4	ß1=0,7063622074 ß2=0,1968711449 ß3=-15,38040005	94,98%	51,98%	ß1=-0,1015740974 ß2=0,02780071312 ß3=-9,960160522	96,65%	96,03%
62,1	ß1=0,6328939257 ß2=0,1659178004 ß3=-13,42700919	93,97%	51,65%	ß1=-0,02996968424 ß2=0,02100088093 ß3=-8,148386616	94,64%	94,37%
103,5	ß1=0,3688207096 ß2=0,0887492657 ß3=-7,478607445	90,63%	53,31%	ß1=0,0119868636 ß2=0,01011543193 ß3=-4,245514698	91,30%	89,40%

	Le modèle probit Y=ß1*x1+ß1*x1*x2+e Les variables exogènes :X1;X2			Le réseau de neurones Y=ß1*x1+ß1*x1*x2+e Les variables exogènes :X1;X1*X2
La variance du bruit e:N(0,si)	Les coefficients estimés	Le taux de succés sur les données de calibration	Le taux de succés sur les données de test	Le taux de succés sur les données de test
0	ß1=0,1906730278 ß2=-0,04997308988	67,55%	70,43%	95,33%
20,7	ß1=0,1752899163 ß2=-0,04389194752	64,88%	71,00%	91,64%
41,4	ß1=0,1725771473 ß2=-0,04274028618	64,00%	70,00%	84,00%
62,1	ß1=0,1806207902 ß2=-0,04578429144	66,22%	70,43%	86,00%
103,5	ß1=0,1801312077 ß2=-0,04583215648	65,88%	66,55%	74,62%

Figure 6: Comparaison des résultats numériques du Chapitre 5

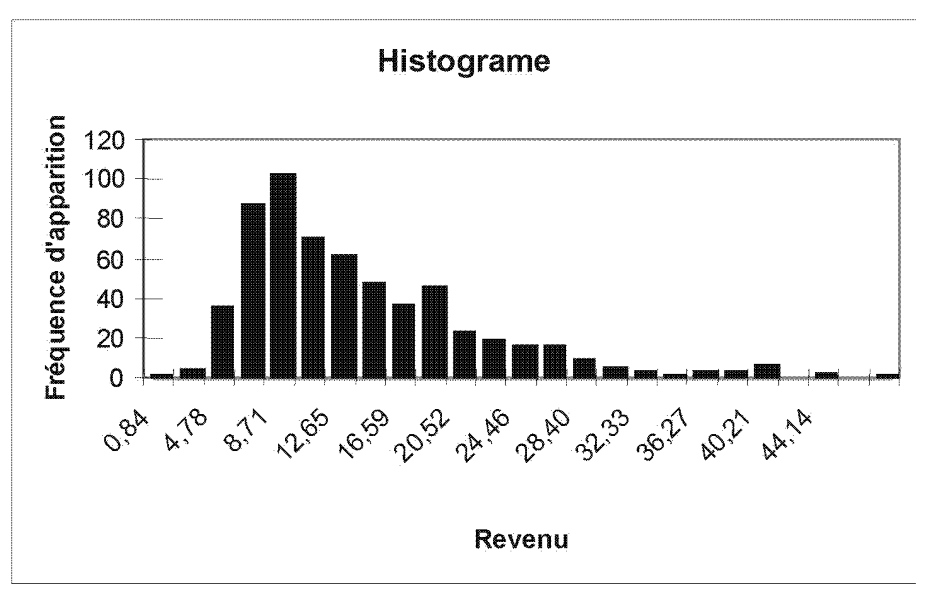


Figure 7: L'histograme des revenus d'un échantillon de 600 individus américains. Source: Ruud (2000), v. Annexe

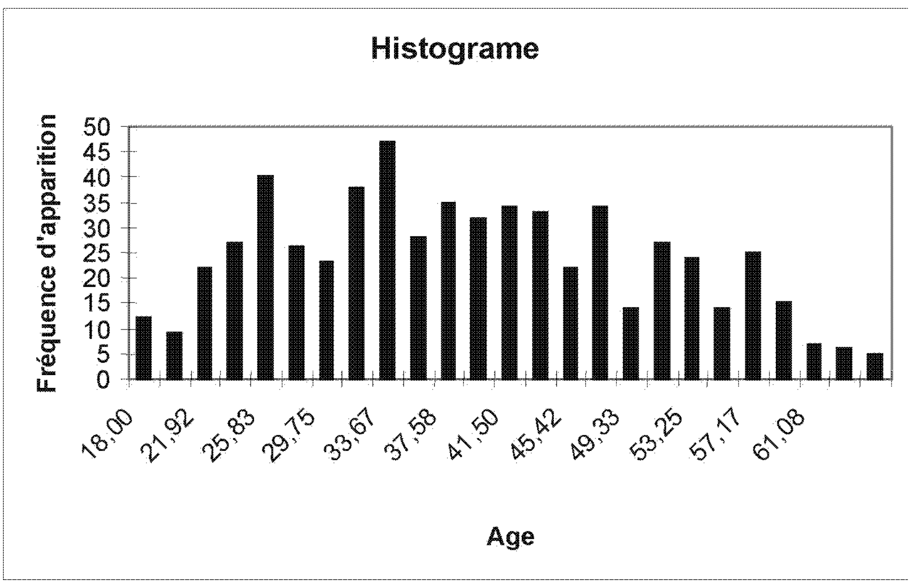


Figure 8: L'histograme des ages d'un échantillon de 600 individus américains. Source: Ruud (2000), v. Annexe

References

[1] A. Ainslie, X. Dreze, *Data Mining: Using Neural Networks as a Benchmark for Model Building*, Décision Marketing, 1996, janvier-avril, pages 9-12.

[2] H.-J. Andreß, J. A. Hagenaars, S. Kühnel, *Analyse von Tabellen und kategorialen Daten,* Springer, 1997.

[3] B. H. Baltagi, *Econometrics*, Springer, 1998.

[4] W. H. Greene, *Econometric analysis*, Prentice-Hall, 1993.

[5] C. Hagen, *Neuronale Netze zur statistischen Datenanalyse*, Shaker, 1997.

[6] A. Monfort, *Statistique*, Univ. Ecole Polytechnique, 2000.

[7] G.Nakhaeizadeh, *Data Mining, Theoretische Aspekte und Anwendungen*, Physica, 1998.

[8] P. A. Ruud, *An Introduction to Classical Econometric Theory*, Current Population Survey, March 1995, U.S. Bureau of the Census, Oxford University Press, 2000.

[9] S. Sarle, *Neural Networks and Statistical Models*, Proceeding of Nineteenth Annual SAS Users Group International Conference, avril 1994, pages 3-6.

[10] A. Scherer, *Neuronale Netze: Grundlagen und Anwendungen*, Vieweg, 1997.

[11] H. Theil, *Principles of Econometrics*, John Wiley & Sons, 1971.

[12] K.-P. Wiedmann, F. Buckler, *Neuronale Netze im Management.* In: Wiedmann/Buckler (Hrsg.): Neuronale Netze im Marketing-Management, Praxisorientierte Einführung in modernes Data-Mining, Gabler, 2001, pages 15-34.

[13] K.-P. Wiedmann, F. Buckler, H. Buxel, *Data Mining: Ein einführender Überblick*, In: Wiedmann/Buckler (Hrsg.): Neuronale Netze im Marketing-Management, Praxisorientierte Einführung in modernes Data-Mining, Gabler, 2001, pages 37-100.

[14] L. Zeng, *Prediction and Classification with Neural Network Models*, Prepared for delivery at the American Political Science Association Annual Meeting, San Francisco, 1996.

[15] J. Hertz, A. Krogh, R. G. Palmer, *Introduction to the Theory of Neural Computation,* Addison Wesley, 1991.

Nicoleta Minoiu
Université " Politehnica " Bucarest,
Département de Sciences de l'Ingénieur, filière allemande,
Etudiante 5e année, diplôme d'ingénieur économiste à partir de septembre 2002
Alfred Messel Weg 10 A-62 64289 Darmstadt, Allemagne,
E-mail: minoiunicoleta@yahoo.de